

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re application of : 22 May 2000  
Alessandro Donatelli et al :  
Serial No.: : IBM Corporation - T81/062  
: P.O. Box 12195  
: Research Triangle Park,  
Filed: concurrently herewith : North Carolina 27709  
:   
For: Managing Pervasive Devices : Group Art Unit:  
: Examiner:

**Preliminary Amendment**

Assistant Commissioner for Patents  
Washington, DC 20231

Sir:

Please amend the application filed concurrently herewith as follows:

**In the Specification:**

Page 1, line 8, change "software distribution and enterprise" to -Software Distribution Version 3.6.-;

Page 1, line 9, change "system management, and detailed" to -Detailed-;

Page 1, line 9, change "Tivoli V3.6" to -this product-;

Page 1, line 10, change "at <http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg242045.pdf>" to -in New Features in Tivoli Software Distribution from International Business Machines Corporation, Part #SG24-2045-0, ISBN 0738401099-;

Page 1, line 13, after "ORB" insert -(Object Request Broker)-;

Page 1, line 13, after -18"-, insert -connection-;

Serial No.:

-1-

Docket No: GB920000048US1

Page 2, line 16, between "although" and "others", delete the comma;

Page 4, between lines 9 and 10, insert **-Detailed Description-**;

Page 5, line 11, delete "Tivoli";

Page 5, line 22, delete "Description of the Preferred Embodiments";

Page 6, lines 5, 6, 8, 14, delete each occurrence of "Tivoli";

Page 6, line 7, delete "Tivoli's";

Page 9, lines 21 and 24, delete each occurrence of "Tivoli";

Page 10, lines 1, 6, 8, 9, 22, 25 and 26, delete each occurrence of "Tivoli";

Page 11, line 10, add commas after "to" and "turn";

Page 13, line 22, delete "Tivoli";

### **In the Claims:**

Please **CANCEL** claims 1 through 16 without prejudice and replace them with claims 17 through 32 appearing in Attachment A to this preliminary amendment.

### **REMARKS**

The specification of the subject application has been amended to eliminate typographical informalities. The original claims have been replaced with new claims that more clearly comply with the requirements of United States practice.

Respectfully submitted,



Gerald R. Woods  
Attorney of Record  
Registration No. 24,144

Docket: GB920000048US1

Telephone: (919) 7204

Fax: (919) 254 - 4330

## Attachment A

1 17.(new) For managing pervasive devices, a gateway component resident on a workstation, said  
2 gateway component being instantiable during synchronisation of said workstation with a  
3 pervasive device and comprising:

4 means for transferring a device agent to a pervasive device; and

5 means for transmitting configuration information to the device agent, said agent  
6 comprising means for executing configuration commands in response to the configuration  
7 information received from the gateway component.

1 18.(new) A gateway component as defined in claim 17 further including:

2 means for receiving a file from a management server including the address of a specific  
3 pervasive device and one or more commands;

4 means for generating device-specific commands based on the received file; and

5 means for forwarding the device-specific commands to the device agent at the specific  
6 pervasive device identified in the file received from the management server, said device  
7 agent executing the device-specific commands as they are received.

1 19.(new) A gateway component as identified in claim 18 wherein the commands comprise  
2 commands for removing files from the specific pervasive device.

1 20.(new) A gateway component as identified in claim 17 further including:

2 means for receiving a file from a management server including the address of a specific

3 pervasive device and one or more commands;

4 means for generating device-specific commands based on the received file; and

5 means for forwarding the device-specific commands to the device agent at the specific  
6 pervasive device identified in the file received from the management server, said device  
7 agent storing the device-specific commands for execution after all are received.

1 21.(new) A gateway component as identified in claim 20 wherein the commands comprising  
2 database or application configuration commands.

1 22.(new) A system for managing pervasive devices,

2 a gateway component resident on a workstation, said gateway component being instantiable  
3 during synchronisation of said workstation with a pervasive device and comprising:

4 means for transferring a device agent to a pervasive device, and

5 means for transmitting configuration information to the device agent; and

6 a pervasive device component including the device agent received from the workstation, said  
7 device agent including means for executing configuration commands in response to the  
8 configuration information received from the gateway component.

1 23.(new) A system as defined in claim 22 wherein the device agent in said pervasive device  
2 component includes means for deleting the configuration commands when the pervasive device  
3 has been configured.

1 24.(new) A system as defined in claim 23 wherein said pervasive device component includes

means for deleting the device agent once configuration is complete.

25.(new) A system as defined in claim 22 further including:

a controller resident on the workstation for pervasive devices of a given type, said controller instantiating one or more modules during synchronisation of devices of the given type; and

an enabling component including means for configuring the controller to add said gateway component as a module to any modules instantiated during synchronisation of pervasive devices of the given type.

26.(new) A system as defined in claim 25 wherein said pervasive device is a Palm Computing Platform device and wherein said controller comprises a mask defining any conduit modules which are instantiated during synchronisation of a pervasive device and wherein said enabling component comprises means for configuring said controller to selectively add said gateway component as a module to any modules which are instantiated during synchronisation of said pervasive device.

27.(new) A system as defined in claim 22 wherein said device agent, in response to a request from said gateway agent, performs an inventory of software installed on the pervasive device and returns the inventory to said gateway component.

28.(new) A method for managing pervasive devices comprising the steps of:

instantiating a gateway component to be resident on a workstation during synchronisation of said workstation with a pervasive device,

transferring a device agent from the gateway component to the pervasive device;

transmitting configuration information from the gateway component to the device agent at the pervasive device, and

executing configuration commands at the pervasive device in response to configuration information received from the gateway component.

29.(new) A method as defined in claim 28 including the additional step of deleting the configuration commands at the pervasive device when the pervasive device has been configured.

30.(new) A system as defined in claim 29 including the additional step of deleting the device agent at the pervasive device once configuration is complete.

31.(new) A system as defined in claims 28 - 30 including the additional step of having the device agent at the pervasive device perform an inventory of software installed on the pervasive device and return the inventory to said gateway component.

32.(new) A program product comprising a computer usable medium having a computer readable program embodied in said medium, wherein the computer readable program when executed on a computer causes the computer to:

instantiate a gateway component resident on a workstation during synchronisation of said workstation with a pervasive device,

transfer a device agent from the gateway component to the pervasive device;

transmit configuration information from the gateway component to the device agent at the pervasive device, and

direct execution of configuration commands at the pervasive device following receipt of

configuration information from the gateway component.

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

# Managing Pervasive Devices

## Field of the Invention

The present invention relates to a management system for pervasive devices.

## Background of the Invention

A brief overview of a prior art three tier management system is shown in Figure 1. Here, a management server 10 communicates with management agent 12 resident on an endpoint computer workstation 14 (only one shown) via a gateway 16 (only one shown).

A well known example of such a system is Tivoli software distribution and enterprise systems management, and detailed information about the features offered by Tivoli V3.6 are available at <http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg242045.pdf>.

In a Tivoli system, each enabled endpoint 14 communicates with a gateway 16 via any peer-to-peer protocol, for example, TCP/IP, IPX or SNA. Communication between the server 10 and the gateway 16 is via an ORB 18', 18'', where each endpoint connected to a gateway is addressed by the server through an associated object on the gateway ORB 18'', thus enabling the endpoint to be directly addressed by the server. A management agent 12 known as a Lightweight Client Framework (LCF), resident on the endpoint, is thus accessible by the server and enables the server to push software via the gateway to the endpoint and call methods on applications resident on the endpoint. The server in turn runs programs 20, 22, 24 allowing: software to be selected for distribution to specified endpoints or defined groups of endpoints; endpoint software inventories to be obtained; and endpoint activity to be monitored via an Event Console, and these programs are extended for each endpoint platform to be managed.



1 While this management system has been quite successful, it should be noted that in order  
2 to control any endpoint 14, the endpoint must itself be manually enabled by installing the LCF  
3 client 12 on the endpoint and configuring the LCF client accordingly, for example, setting up its  
4 gateway address and communication protocol.

5 At the same time more and more users are employing Tier 0 or pervasive devices  
6 ("devices") 26 including, for example, Palm Computing® platform devices ("Palm devices")  
7 from Palm Inc., a range of organizers from Psion plc and to a growing extent mobile and smart  
8 phones. Typically data on such pervasive devices 26 is managed locally by periodically  
9 connecting the pervasive device to a workstation 12 running a controller program 28 and  
10 synchronising one or more databases stored on the device, for example an address book, with  
11 corresponding databases stored on the workstation.

12 In the case of a Palm device, a workstation application called HotSync® Manager  
13 controls the synchronisation of databases as well as the installation of new applications. HotSync  
14 uses one or more plug-ins known as conduits, each for exchanging and synchronising data  
15 between the workstation 14 and the Palm device 26. Most conduits synchronise data such that  
16 data on the device mirrors the data on the workstation, although, others also transfer,  
17 import/export data, or cause Palm applications to be installed.

18 One of the default conduit modules (netcond.dll) responds to a user placing a Palm  
19 Network Configuration (.PNC) file in a pre-determined sub-directory prior to synchronisation.  
20 The file is copied to the device by the conduit module where it is used to update the script file for  
21 a modem. This, however, is extremely limited in terms of aspects of the device that may be  
22 configured. Also, the Palm desktop 28 includes a set-up menu, but this only enables a user to set  
23 workstation side modem and network parameters. If these are at odds with the Palm device, then  
24 errors can occur.

1 Because configuration facilities are so limited, perhaps because of the problems of trying  
2 to access databases on the pervasive device which might be in use and so locked or unavailable,  
3 thus causing such facilities to operate unreliably, new users of pervasive devices may end up  
4 spending more time than is necessary manually configuring the devices before using them  
5 profitably. Even the provision of a configuration program per se is not as helpful as it might be,  
6 because if it is to avoid the problem of accessing databases which might be locked during  
7 synchronisation, it needs to run separately to device synchronisation and thus places an extra  
8 burden on the user. This also make the management of such devices within a tiered structure  
9 more difficult.

10 If such pervasive devices are to be employed more efficiently within an enterprise, then  
11 they need to be quickly and easily configurable as well as capable of receiving and having  
12 enterprise applications and data updated as seamlessly as possible, preferably, within the same  
13 framework used to manage other resources within the enterprise.

#### 14 Summary of the Invention

15 Accordingly the present invention provides a management system comprising a gateway  
16 component adapted to reside on a workstation and a device agent adapted to reside on a  
17 pervasive device for configuring pervasive devices; said gateway component being instantiable  
18 during synchronisation of said workstation with a pervasive device and comprising: means for  
19 transferring said device agent to said pervasive device; and means for transmitting configuration  
20 information to said pervasive device; and said device agent comprising means for executing  
21 configuration commands in response to said configuration information.

22 The invention enables pervasive devices to be managed through a workstation to which  
23 they connect to synchronise without requiring any intervention by the pervasive device user.  
24 Preferably, the workstation acts as a gateway for managing the device within what becomes a  
25 four tier management system.

1     **Brief Description of the Drawings**

2             Embodiments of the invention will now be described with reference to the accompanying  
3     drawings, in which:

4             Figure 1 is a schematic diagram of a prior art three tier management system;

5             Figure 2 is a schematic diagram of a four tier management system for controlling  
6     pervasive devices;

7             Figure 3 is a diagram of data stored on an endpoint through which Tier 0 Palm Pilot  
8     devices are managed according to the invention; and

9             Figure 4 is a flow diagram illustrating the operation of an endpoint management gateway.

10            In the preferred embodiment of the present invention, Figure 2, the prior art management  
11     server is extended to provide respective inventory, software distribution and event console  
12     programs 20', 22', 24' for each pervasive device type to be managed by the system.

13            It can be seen, however, that there are likely to be many more pervasive devices 26 than  
14     the endpoints 14 traditionally managed within such a system and so addressing each such device  
15     26 via a respective object on the gateway ORB 18" would be too onerous. The management  
16     system therefore treats such devices 26 as residing on a fourth tier in the management system,  
17     with each device being managed via an agent 30 resident on a suitably enabled endpoint. Thus,  
18     each enabled endpoint 14 acts as a management gateway for devices.

19            The management server 10' employs a device storage manager 32 which contains the  
20     information about the device types and instances of devices being managed. For example, a  
21     device type of "Handheld Device" contains many instances of this type of device. This allows any

1 management server applications and users to add devices, edit devices, delete devices, and query  
2 for devices. Each such device in the system is identified by 5 pieces of information:

3 1. Device Type

4 2. ID - a unique identifier assigned by the system to each device

5 3. Label - a friendly name for a device, specified at creation time. A Type plus a Label  
6 should uniquely define a device.

7 4. Manager - the identity of an endpoint which is to act as a management gateway for this  
8 device.

9 5. Local Address - the local address of a device in the context of its management gateway

10 Manager is the gateway object reference through which to route management operations  
11 (using Tivoli method calls) toward a given device, thus the server-gateway ORB 18',18" need  
12 only handle the same number of object references as within a three tier management system. It  
13 should therefore be seen that, even though the number and level of devices being managed within  
14 the system of the preferred embodiment is magnified, the use of a single ORB object not only to  
15 address the endpoint with which it was traditionally associated, but to manage essentially a  
16 hierarchy including an endpoint and a plurality of devices, prevents the endpoint addressing  
17 mechanism from collapsing from the sheer number of devices being controlled.

18 Local Address is an address for a device at least unique in the context of its management  
19 gateway. So, using this information, it is possible to locate and manage any device in the  
20 management system. This scheme also allows for every device to have a unique ID and a unique  
21 friendly name (Label).

## 22 Description of the Preferred Embodiments

23 In the preferred embodiment, the device storage manager 32 encapsulates how the device  
24 data is stored and managed. Thus, the device storage manager and other applications such as the

1 programs 20', 22' and 24' only need to agree on a data exchange protocol so that messages can  
2 be exchanged. Using this mechanism, the management server 10' is able to support arbitrary  
3 Storage Manager implementations (e.g. disk files, LDAP servers, RDBMS)

4 Furthermore, the management server 10' provides a device grouping mechanism which is  
5 a Tivoli managed resource capable of managing groups of devices. A device group can be a  
6 subscriber to a conventional Tivoli Profile Manager. This enables devices to be managed via  
7 Tivoli's management by subscription paradigm. The device group also implements policy and  
8 security for devices. Device groups can thus be made managed resources of Tivoli Policy  
9 Regions, and so it is possible to create custom policies for device group instances and to enforce  
10 security models for devices through the Policy Region mechanism. By granting access to device  
11 groups on a per Policy Region basis, administrators may be selectively granted access to create  
12 device groups, manage subscriptions to device groups, and to manipulate device groups  
13 distributions. This depends on roles each administrator holds in each Policy Region and is  
14 normal Tivoli policy and security.

15 So, using the above methodology, the management system is able to identify endpoints 14  
16 which are to act as management gateways for pervasive devices 26, and the management of such  
17 devices will now be discussed in relation to Palm devices in particular.

18 In order to enable endpoints as management agents for Palm devices, the endpoint 14  
19 must first have a Palm desktop 28 installed. This can be done either manually on the workstation  
20 14 or through the management server 10' deploying and running the Palm desktop installation  
21 software in a conventional manner. During installation of the Palm desktop on a Windows  
22 platform, certain changes are made to the Windows registry, Figure 3, in particular, to the  
23 HotSync Manager branch of the registry which includes two default conduits. Each conduit  
24 branch on an endpoint is given a name "Conduit x", where x is an ordinal number unique on the  
25 endpoint. Each conduit branch includes, inter alia, the following set of variables:

1 Module, which tells HotSync which DLL program is associated with the conduit;  
2 Extensions which conventionally indicates to the DLL which file types it is to synchronise  
3 with the Palm device;  
4  
5 Directory tells the DLL the name of the device sub-directory in which to find the files to  
6 synchronise; and  
7  
8 Creator ID which a special four character string associated with each conduit program.

8 Conduit 1, is the install conduit and installs/synchronises Palm applications (.prc files)  
9 and databases (.pdb files) stored in an "install" sub-directory for a particular device. Every .pdb  
10 database requires the .prc application that will access that database type. Where Palm VII devices  
11 and 3.2 Palm OS are being managed then Palm Query Applications (.pqa) files are also handled  
12 by the install conduit.

13 Conduit 2, operates on Palm Network Configuration Files (\*.pnc) also called Palm  
14 Network Script Files (\*.scp) stored in a "TCP" sub-directory for a particular device. The conduit  
15 module interprets the lines contained in the .pnc and .scp files to execute on the device a  
16 restricted set of network operations like: "wait for", "Send", "Send CR", "Send User ID", "Send  
17 Password", "Delay", "End".

18 Once the desktop controller 28 is installed, an enabling program 34 and a management  
19 agent 30, TMPP.DLL, can be injected into the endpoint 14. The enabling program adds a further  
20 conduit branch to the registry. The program 34 detects installed conduits and if, as in the present  
21 example, only the default conduits are installed, the new conduit's name is set to "Conduit 4".  
22 This conduit's module is set to be the management agent 30, with a creator ID of "TIVO", which  
23 operates on files of extension. PCF stored in a "Tivoli" sub-directory for a device, Figure 3.

1 At the same time users may already be connecting and synchronising unmanaged devices  
2 through the Palm desktop. When a device connects to an endpoint, it sets up a conduit mask  
3 defining which conduits are to execute when the device synchronises. Each conduit mask for a  
4 device comprises a Windows registry variable named "Installxxxx", whose value determines  
5 which of the defined conduits are to execute when HotSync executes for a device. The string  
6 xxxx is linked with the device directory name, from which the Install, TCP and Tivoli  
7 sub-directories depend, through the file Users.dat. So once a device is synchronising with an  
8 endpoint, the association of the conduit mask and the device directory name is maintained by the  
9 Palm Desktop and the Palm Desktop API exposes this relationship. This enables any program,  
10 such as the Palm Desktop itself or modules such as TMPP.DLL to a) locate the conduit mask for  
11 a device; b) locate the sub-directory in which Palm files are stored for a device being  
12 synchronised; and c) determine the names of any devices (managed or unmanaged) connecting to  
13 the endpoint.

14 At the management server 10', an enabled endpoint which is to act or acts as a  
15 management agent for a device is either directly identified by a management server user, or the  
16 user can request specific endpoints or groups of endpoints to return the identities of devices that  
17 a) they manage; or b) connect to the endpoint, even if they are not currently managed. In the  
18 preferred embodiment, these identities comprise a triplet of the form <label><manager><Palm  
19 Desktop ID>. (The Palm Desktop ID comprises the Windows user ID of the owner of the Palm  
20 Desktop managing the device).

21 Thus, the management server can seek out the identities of devices connected to enabled  
22 endpoints, add the identities of any unmanaged devices to the device storage manager 32 and  
23 then configure the associated endpoints to manage these devices. This configuration comprises  
24 updating the conduit mask for the device on the endpoint via the program 34 or by any other  
25 program injected into the endpoint, to cause the Tivoli conduit to execute during synchronisation.

1           So it will be seen that without any intervention either on the device or on its associated  
2 endpoint, other than the user simply synchronising their device as normal, the endpoint has been  
3 set up to manage the device from the server in a No-Touch manner. The types of management  
4 that can be carried out on the device correspond to those normally carried out on managed  
5 endpoints, that is, software distribution, inventory and event monitoring. In addition, the  
6 preferred embodiment, enables the management server to configure individual or groups of  
7 devices and to remove databases/applications from devices.

8           Once a device 26 is associated with such an endpoint 14 which is to act as a management  
9 gateway, the simplest aspect of remotely and non-interactively configuring a Palm device via a  
10 tiered management structure is to distribute software applications. To deploy applications to the  
11 Palm device, the management server software distribution program 20' simply causes the  
12 appropriate .prc, .pdb and/or .pqa files to be placed in the Install directory for a device of group  
13 of devices. To do this, application files are supplied by the management server to an identified  
14 endpoint. An endpoint program in turn locates the appropriate Install sub-directory name for the  
15 device label supplied (again using the Users.dat information accessed via the Palm desktop API)  
16 and places the files in the sub-directory. When synchronisation next takes place with the device,  
17 the install conduit copies these files from the device's Install sub-directory on the endpoint to the  
18 device. It will be seen that this step requires no extra effort or intervention by the user of either  
19 the endpoint or the Palm device and so ensures that applications will be deployed to the Palm  
20 device at the very next opportunity, that is, the next synchronisation.

21           The Tivoli conduit module 30 acting as the management gateway for the device also  
22 performs the functions of carrying out a software inventory, notifying events, configuring the  
23 Palm device and removing applications from the Palm device.

24           In the present embodiment, the Tivoli conduit module TMPP.DLL is instantiated during  
25 synchronisation of a device, step 40, Figure 4. The module first checks the device to determine if  
26 an up-to-date copy of an agent Tivoli.prc 38 is available on the device. If not an up-to-date agent



1 is copied to the Palm device, step 42. It should be noted that the Tivoli conduit module does not  
2 rely on the Install conduit having copied Tivoli.prc to the device, as this would only make  
3 Tivoli.prc accessible to the install conduit module during that synchronisation cycle. If this  
4 approach were adopted, then either a second synchronisation cycle would be required or the  
5 install conduit module would need to be adapted to carry out the configuration otherwise  
6 performed by the Tivoli conduit module. Nonetheless, should this restriction be lifted or not  
7 pertain for other device types, it would be possible to place the agent on the device using the  
8 install conduit and for the Tivoli conduit module to communicate with the agent thereafter.

9 In any case, in the preferred embodiment, the Tivoli conduit module 30 then processes  
10 any device configuration information, step 44, previously supplied from the management server  
11 10' through the LCF 12 in the form of a .pcf file and placed in the Tivoli sub-directory for the  
12 device, step 46. It will be seen, however, that the configuration information can be supplied from  
13 any source, for example, a desktop user generated script file or via a desktop controller GUI. It  
14 should be noted that the conduit module can read from write locked device databases during  
15 synchronisation and this may be required to ascertain the version and thus the data structure of  
16 databases which are to be updated. Thus, during step 44, the conduit module can use the known  
17 structure of a database to be configured, to convert the configuration information into Palm  
18 device commands which it then writes to a device database Tivoli.pdb 39 again using the Palm  
19 desktop API, step 48. The conversion into Palm commands is done within the endpoint 14 as this  
20 both reduces the footprint required on the device and enables the PCF file to be written for more  
21 than one device independent of the versions of its databases. During synchronisation, the conduit  
22 module sets an alarm condition, step 50. The condition specifies the Tivoli agent service the  
23 alarm with the alarm passing to the agent 38 a launch code parameter indicating that it is to  
24 service an alarm.

25 In the case of an alarm, the Tivoli agent executes the commands stored in the temporary  
26 database 39. Thus, when the Tivoli conduit module signals that synchronisation is complete, step  
27 52, the databases locked during synchronisation are released, and so the database updating

1 commands stored in the temporary database can be subsequently executed properly by the agent,  
2 step 54. Because the configuration commands execute automatically and immediately after  
3 synchronisation, it is also likely that the Palm device user will not be doing anything which is  
4 likely to lock a database, thus making configuration more likely to succeed.

5 The Palm commands generated generally comprise a sequence of: open database, write  
6 value, close database, although some configuration may result in single commands such as set  
7 network preference. In more detail, however, the syntax definition for the .pcf file is defined in  
8 Appendix A. (Some sections of the definition have been foreshortened for clarity with the deleted  
9 portions indicated by ellipsis.) This definition is stored in a local Tivoli.DEF file which is read  
10 and parsed by the conduit module, during step 44, thus enabling it to in turn parse and process  
11 the .pcf file. While the current definition is written to a bespoke grammar, it will be seen that the  
12 definition could be written in any suitable language, for example, Extended Mark-up Language,  
13 which in turn could be parsed by an off-the-shelf parser incorporated in the conduit module.  
14 Alternatively, the definition could be built into the conduit module, although this would make  
15 maintenance more difficult.

16 It can be seen from the definition file that ";" "\*" and "!" are reserved characters, whose  
17 function is explained below, and that text following pound (#) is ignored. According to the  
18 syntax definition, .pcf files are divided into stanzas, each beginning with a label in square  
19 brackets, for example [SYSTEM]. Within each stanza, each line comprises an expression of the  
20 form "preference name = value". The conduit module associates each stanza label with either a  
21 database or a type of command, for example, [System] is associated with the "Saved preferences"  
22 database and [Network] is associated with the command "Set network preference".

23 Some databases, such as "Saved preferences" have multiple known versions each with a  
24 respective data structure. When the conduit module reads a stanza label for such a database, it  
25 tries to identify which version and thus which data structure is used by the database. In the case  
26 of "Saved preferences", each version is stored in the first two bytes of the database. The current

1 syntax definition for SYSTEM allows the conduit module to operate with versions 3 to 7 of the  
2 Saved preferences database. For each version, the definition includes an associated sysStruct,  
3 telling the conduit module where variables are stored within the database. Each sysStruct element  
4 comprises a list of <preference name><offset><size> triplets, each separated by "|" char. A  
5 negative value in <size> means a bit position at the specified offset. So, for example, when the  
6 conduit module reads from a database, it knows where to look for the version information, and  
7 can determine, for example, that version 5 of Saved Preferences is being used. The conduit then  
8 knows that, for example, "dateformat" can be set by writing the 3rd byte of the database to the  
9 required value.

10 Each preference name can include a "\*" character, and the conduit module is hard coded  
11 to recognise the string after the "\*" letting it understand that special processing is required for  
12 that preference.

13 Possible values are separated by a semicolon, so for example, the first possible value for  
14 country is Australia and the next Austria. So, if a .pcf file included the following stanza:

15 [SYSTEM]

16 country = Austria

17 and the Palm device were using version 5 of Saved Preferences, then the conduit would send the  
18 following (paraphrased) commands to the Tivoli Agent for storage in the temporary database 39:

19 Open "Saved Preferences"

20 Write 1 to location 2

21 Close Database

22 Each value can have aliases which are separated by "pipe" (|) char. So for example, the  
23 conduit will recognise that if "UnitedKingdom" or "United Kingdom" is specified, the same

country value will be written in Saved Preferences. Also, a "\*" char preceding a value means that it is the default value for that preference name. It means that a statement in the pcf file can indicate "preference name=default" and the value with the "\*" will be set on the preference name.

Finally, some lists of values start with a ";" char because the internal representation of the preference name starts counting from 1 rather than 0. The ";" char can thus be inserted or deleted from the beginning of the value list for a preference value in the definition file, to shift values between those set on the Palm device and the configured ones in the pcf file.

Once the temporary database of configuration commands has executed, step 54, the agent finishes by deleting the temporary database and then optionally deleting itself, so removing any management system presence on the Palm device.

Another task of the conduit module is to cause applications and databases to be removed from the device. In the preferred embodiment, the management server places a .rmv file in the Tivoli sub-directory for the device. During synchronisation, the conduit module looks for such a file and, if found, reads the file to identify the database/applications to be removed. It then creates the appropriate Palm device commands and executes these, again through the Palm desktop API, step 58. The database/application remove commands are executed during synchronisation, in spite of the fact that the database/application may be locked. This is considered beneficial as if it is not possible to delete such a database/application, then the conduit module can in turn report back to the management server Event Console. This should cause the management server user to think about why they wish to delete a database/application which is in fact in use.

In the preferred embodiment, the Tivoli device agent 38 can also be called during synchronisation. If the server program 22' requires an inventory of software on the device to be taken, then the agent is responsive to the conduit module 30 calling the agent 38 with a different launch code than that used for serving the alarm condition, step 56. The agent responds to this

1 inventory launch code, prepares an inventory and returns this to the workstation, from where it is  
2 returned to the server 10' in a conventional manner.

3 The conduit module can also be extended notify any synchronisation events, such as the  
4 successful deployment of the database 39 or removal of databases, to the management server  
5 Event Console program 24', step 58, so ensuring that the complete range of management services  
6 for conventional endpoints can also be executed on pervasive devices.

GB920000048US1

## Appendix A

[SYSTEM]

# System preferences

sysStruct[3]=|version,0,2|country,2,1|dateformat,3,1|longdateformat,4,1|weekstartday,5,1|timeformat,6,1|numberformat,7,1|autooffduration,8,1|sysoundlevelv20,9,1|gamesoundlevelv20,10,1|alarmsoundlevelv20,11,1|hidesecretrecords,12,1|devicelocked,13,1|localsyncrequirespassword,14,1|remotesyncrequirespassword,15,1|sysprefflags,16,2|sysbatterykind,18,1|alloweastereggs,19,1|minuteswestofgmt,20,4|daylightsavings,24,2|ronamaticchar,26,2|hard1charappcreator,28,4|hard2charappcreator,32,4|hard3charappcreator,36,4|hard4charappcreator,40,4|calccharappcreator,44,4|hardcradlecharappcreator,48,4|launcherCharAppCreator,52,4|hardcradle2charappcreator,56,4|animationLevel,60,2|DateAndTime,0,0||

sysStruct[4]=|version,0,2|.....|DateAndTime,0,0||

sysStruct[5]=|version,0,2|.....|staylitwhenpluggedin,77,1|DateAndTime,0,0||

sysStruct[6]=|version,0,2|country,2,1|.....|antennacharappcreator,78,4|DateAndTime,0,0||

sysStruct[7]=|version,0,2|country,2,1|.....|measurementsystem,82,2|DateAndTime,0,0||

country\*ctry=Australia;Austria;.....;UnitedKingdom|United Kingdom;UnitedStates|United States;India;Indonesia;Korea;Malaysia;RepChina;Philippines;Singapore;Thailand;Taiwan

dateFormat=MDYWithSlashes|M/D/Y;.....;MYMed|MMM 'YY;MYMedNoPost|MMM YY

longDateFormat=MDYWithSlashes|MM/DD/YY;.....;MYMedNoPost|MMM YY

weekStartDay=Sunday|Sun;Monday|Mon

1 timeFormat=Colon|12HH:MM;.....;Comma|12h|12HH,MM

2 numberFormat=CommaPeriod;PeriodComma;.....;ApostropheComma

3 autoOffDuration=;1 Minute|1|One Minute|1M;.....;3 Minutes|3|Three Minutes|3M

4 sysSoundVolume\*sv1=Off;Low;Medium;High

5 gameSoundVolume\*sv2=Off;Low;Medium;High

6 alarmSoundVolume\*sv3=Off;Low;Medium;High

7 beamReceive\*be=No|False|0|\*Off;Yes|True|1|On

8 stayonwhenpluggedin=No|False|0|\*Off;Yes|True|1|On

9 staylitwhenpluggedin=No|False|0|\*Off;Yes|True|1|On

10 alloweastereggs=No|False|0|\*Off;Yes|True|1|On

11 hidesecretrecords=No|False|0|\*Off;Yes|True|1|On

12 DateAndTime\*dt=\*now|local|PC

13 # Following values for creators must be considered as default and not as a set of possible values.

14 They are CASE SENSITIVE

15 hard1charappcreator\*cr=\*date

16 hard2charappcreator\*cr=\*addr

17 hard3charappcreator\*cr=\*todo

18 hard4charappcreator\*cr=\*memo

19 calccharappcreator\*cr=\*calc

20 hardcradlecharappcreator\*cr=\*sync

21 launcherCharAppCreator\*cr=\*lnch

22 hardcradle2charappcreator\*cr=\*sync

23 measurementSystem=unitsEnglish|inches;unitsMetric|meters

24 [NETWORK]

```

1  # Network Preferences. Note: all the settings for the services are done by pnc files. With the
2  keyword of this section the service can enabled/selected.

3  service*se=*Windows RAS

4  [MODEM]
5  # Modem Preferences
6  modStruct[2]=lversion,0,2lspeed,2,4lspeakervolume,6,1lpulse,7,1lflowcontrol,8,1lresetstring,9,18l
7  initstring,28,8lcountry,0,0lmodemname,0,0ll
8  cDBStruct[0]=lmodemname,0,22lconnectionmethod,22,4lspeed,26,4ldefconf,30,1lisModem,31,1l
9  pulse,32,1lcountry,33,1lspeakervolume,34,2lflowcontrol,36,2lresetstring,38,8linitstring,46,8l||

10 country*co=Argentina;Australia;Austria;.....;UnitedKingdom|United
11 Kingdom;UnitedStates|United States;*other

12 connectionmethod*fvl=*u8EZ
13 isModem=*false|no;true|yes
14 modemname*fvm=*Standard
15 speed*sp=115200;*57600;38400;28800;19200;14400;9600;4800;2400;1200
16 speakervolume=off;*low;medium;high
17 pulse=*touchtone;rotary
18 flowcontrol=*auto;off;on
19 initstring*fvis=*AT&FX4
20 resetstring*fvr=*NC1

21 [HOTSYNC]
22 # Hotsync Preferences

```



```

1 hotmStruct[1]=|version,0,2|pad1,2,1|modemsyncpref,3,1|pad1,4,1|hotsynctype,5,1|localconnectio
2 n,6,22|modemdirectConnection,28,22|modemnetservice,0,0|primarypcname,0,256|primarypcaddress,0,16|prim
3 hotpStruct[2]=|dialprefix,2,-8|disablecallwaiting,2,-7|usecallingcard,2,-6|modemdirectphone,4,41|
4 dialprefixvalue,45,41|disablecallwaitingvalue,86,41|useCallingCardvalue,127,41||

5 modemDirectPhone*fV=*00
6 dialprefix=*false|no;true|yes
7 dialprefixvalue*fV=*9,
8 disablecallwaiting=*false|no;true|yes
9 disablecallwaitingvalue*fV=*1170,
10 usecallingcard=*false|no;true|yes
11 usecallingcardvalue*fV=*,*,*,*
12 modemDirectConnection*fVdc=*Palm V Modem # only PalmOS 3.3 and later
13 modemNetService*fVns=*Windows RAS
14 localConnection*fVlc=*Direct Serial|IR to a PC/Handheld
15 hotsyncType*hsty=*local;modem
16 modemSyncPref=*network;modem
17 LANSyncPref*ls=local|final;LANSync|proxy
18 primaryPcName*fVpn=*localhost
19 primaryPcAddress*fVpa=*127.0.0.1
20 primaryPcSubnetMask*fVps=*255.255.255.0

21 [AUTOPALM]

22 makeimage*pwmi=*h:/Palm/image
23 pcfname*pwpn=*_delta_.pcf
24 # if a path isn't specified, the file is created in image dir
25 makepcf*pwmp=*h:/Palm/image
26 set*pwsb=0:0:0@0,0 #Card:DB:Rec@Offset,byte,byte@Offset,byte....

```

**What is claimed is:**

1. A management system comprising a gateway component adapted to reside on a workstation and a device agent adapted to reside on a pervasive device for configuring pervasive devices;

said gateway component being instantiable during synchronisation of said workstation with a pervasive device and comprising:

means for transferring said device agent to said pervasive device; and

means for transmitting configuration information to said pervasive device; and

said device agent comprising means for executing configuration commands in response to said configuration information.

2. A management system according to claim 1 wherein the configuration information comprises a file including one or more commands.

3. A management system according to claim 2 wherein the file is sent from a management server and addressed to a specific pervasive device.

4. A management system as claimed in claim 2 wherein the gateway component is adapted to generate device specific commands from the file and to transmit them to the device agent when resident on the pervasive device.

5. A management system as claimed in claim 4 wherein the device agent is adapted to execute the commands as they are received.

6. A management system as claimed in claim 5 wherein said commands comprise commands for removing applications or databases from said pervasive device.

1 7. A management system as claimed in claim 2 wherein the gateway component is adapted  
2 to generate device specific commands from the file and to transmit them for storage on the  
3 pervasive device.

4 8. A management system as claimed in claim 7 wherein the device agent is adapted to  
5 execute said stored commands once synchronisation is complete.

6 9. A management system as claimed in claim 8 wherein said commands comprise database  
7 or application configuration commands.

8 10. A management system as claimed in claim 8 wherein the device agent is adapted to delete  
9 said stored commands when configuration is complete.

10 11. A management system as claimed in claim 1 wherein said device agent is adapted to be  
11 deleted from said pervasive device once said configuration is complete.

12 12. A management system as claimed in claim 1 wherein said workstation includes a  
13 controller for pervasive devices of a given type, said controller being adapted to instantiate one or  
14 more modules during synchronisation of devices of said type and wherein said management  
15 system further comprises an enabling component comprising;

16 means for configuring said controller to add said gateway component as a module to any  
17 modules which are instantiated during synchronisation of pervasive devices of said type.

18 13. A management system as claimed in claim 12 wherein said pervasive device is a Palm  
19 Computing Platform device and wherein said controller comprises a mask defining any conduit  
20 modules which are instantiated during synchronisation of a pervasive device and wherein said  
21 enabling component comprises means for configuring said controller to selectively add said

1 gateway component as a module to any modules which are instantiated during synchronisation of  
2 said pervasive device.

3 14. A management system as claimed in claim 1 wherein said management gateway is  
4 adapted to request said device agent, when resident on the device, to perform an inventory of  
5 software installed on the device and to return said inventory to said gateway component.

6 15. A management system as claimed in claim 14 wherein said management gateway is  
7 adapted to return any synchronisation events to a management server.

8 16. A computer program product comprising computer program code stored on a computer  
9 readable storage medium for, when executed on a computing device, managing pervasive  
10 devices, the program code comprising the management system of claim 1.

## ABSTRACT OF THE DISCLOSURE

A management system comprises a gateway component adapted to reside on a workstation and a device agent adapted to reside on a pervasive device for configuring pervasive devices. The gateway component is instantiable during synchronisation of the workstation with a pervasive device to transfer the device agent to the pervasive device; and to transmit configuration information to the pervasive device. The device agent executes configuration commands in response to the configuration information. The invention enables pervasive devices to be managed through a workstation to which they connect to synchronise without requiring any intervention by the pervasive device user. Preferably, the workstation acts as a gateway for managing the device within what becomes a four tier management system.

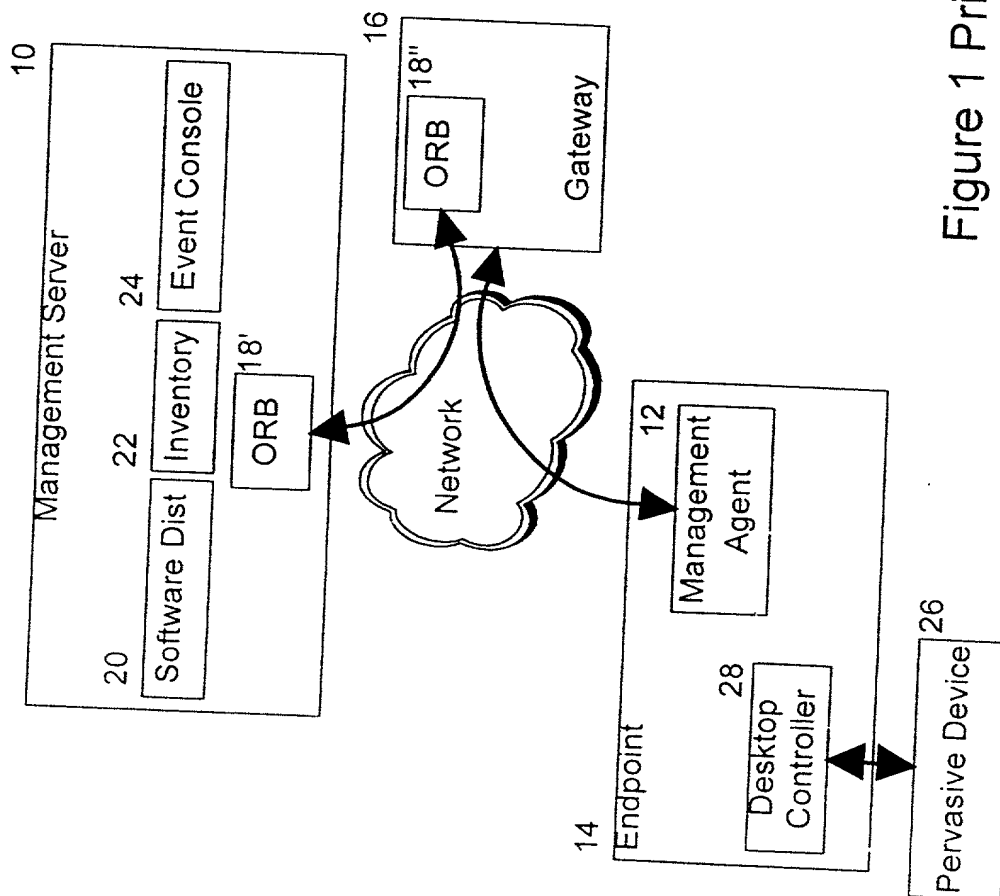


Figure 1 Prior Art

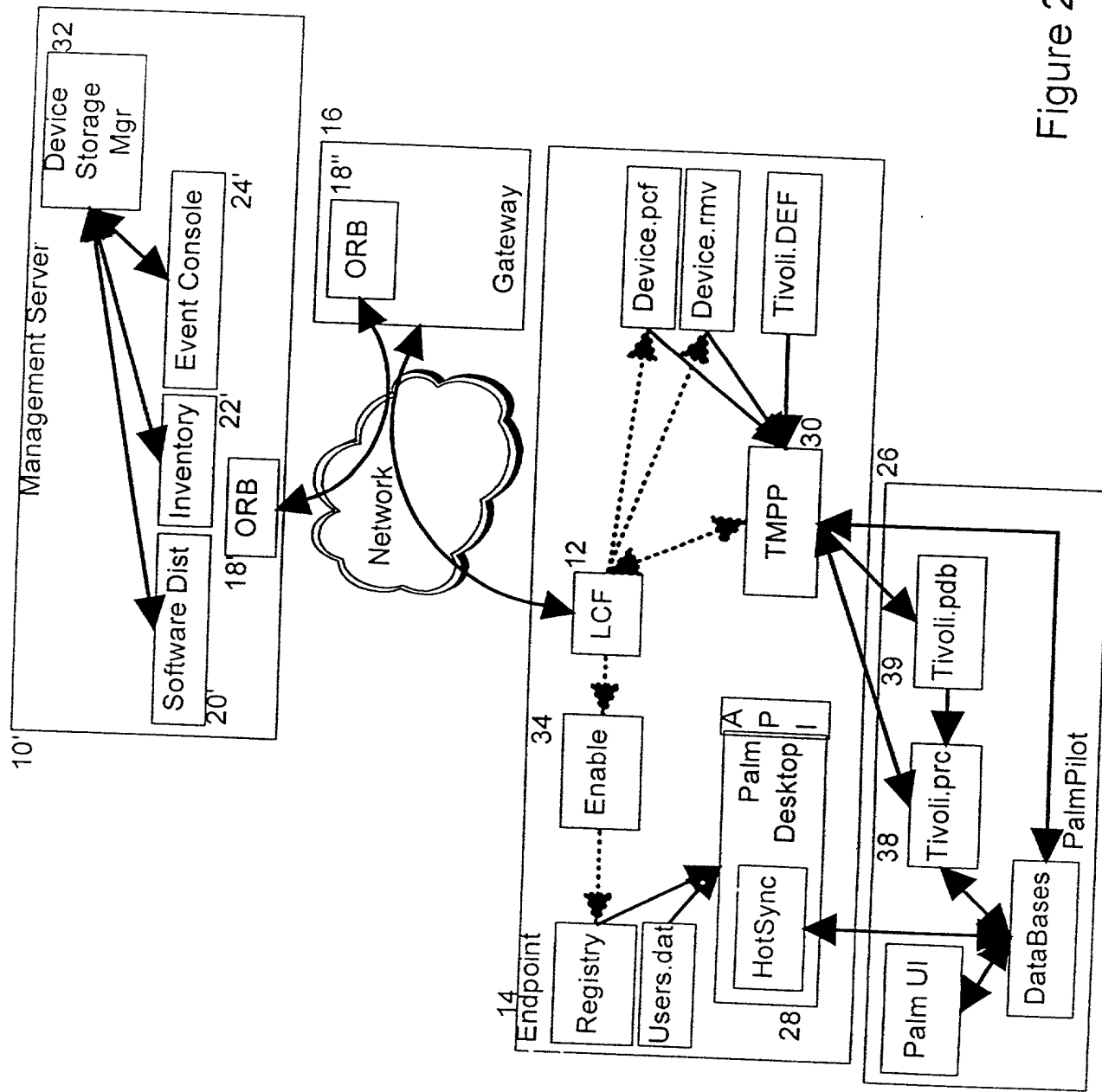


Figure 2

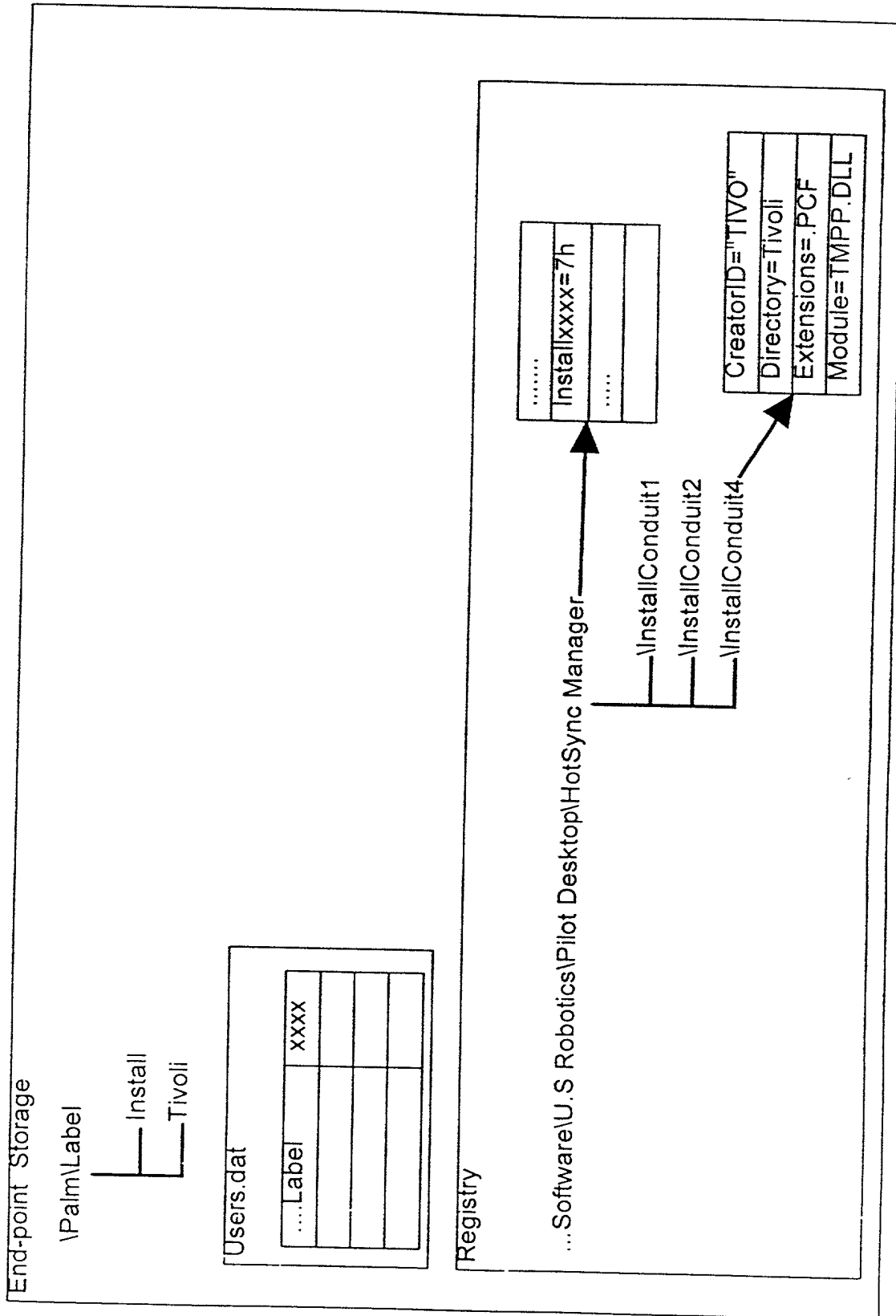


Figure 3



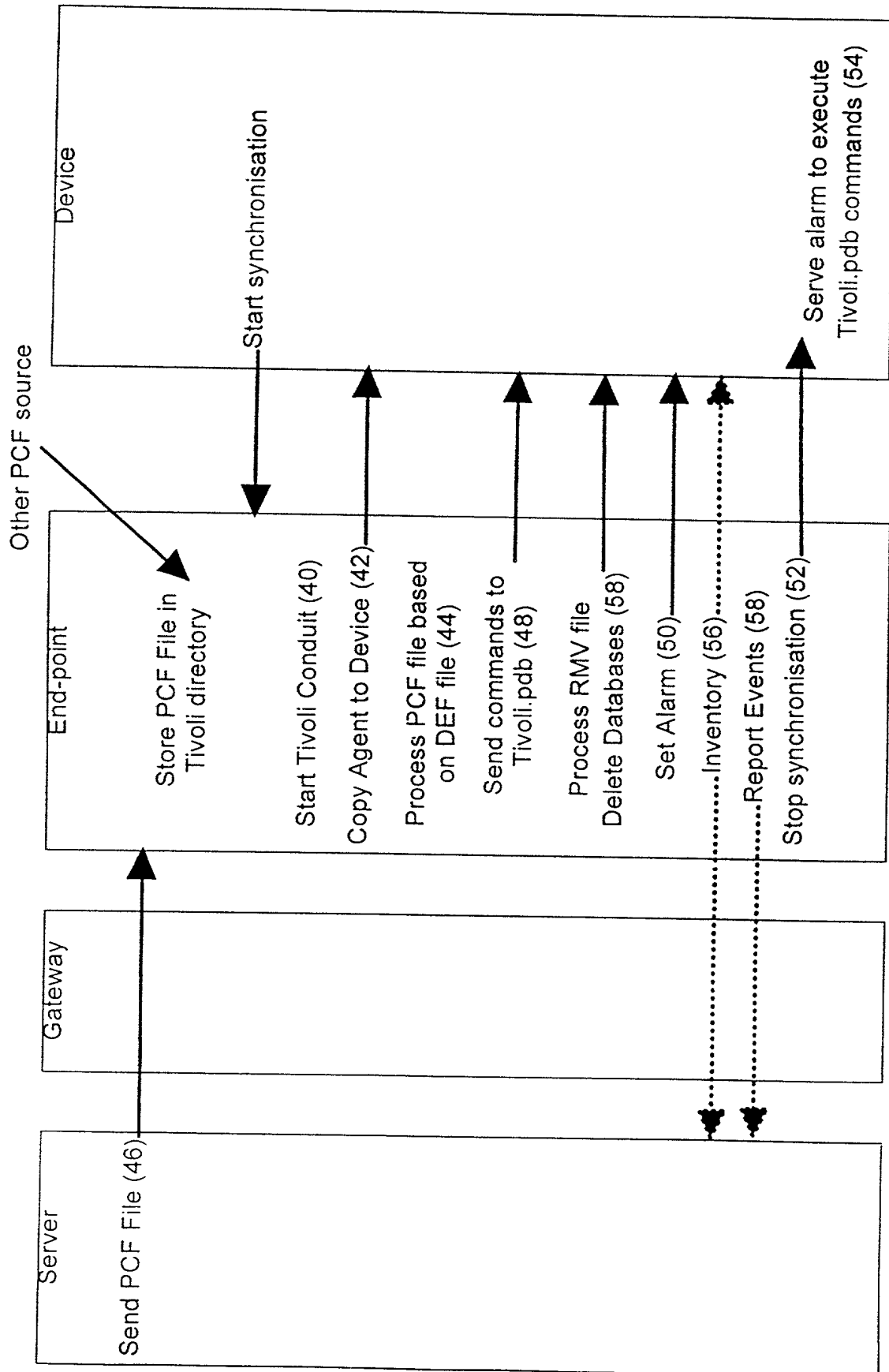


Figure 4

As a below named inventor, I hereby declare that:

## MANAGING PERVASIVE DEVICES

  X   is attached hereto.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, CFR 1.56.

I hereby claim foreign priority benefits under Title 35, USC 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

0006055.8	United Kingdom	14 March 2000	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Number	Country	Filing Date Day/Month/Year	Priority Claimed

I hereby claim the benefit under Title 35, USC 120 of any United States application(s) listed below and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, USC 112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, CFR 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications:

Serial No.	Filing Date	Status
------------	-------------	--------

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Edward H. Duffield, Reg. No. 25,970; A. Bruce Clay, Reg. No. 32,121; Gregory M. Doudnikoff, Reg. No. 32,847; Jeanine S. Ray-Yarletts, Reg. No. 39,808; Jerry W. Herndon, Reg. No. 27,901, Kevin J. Fournier, Reg. No. 34,333

Send all correspondence to:

IBM Corp, IP Law Dept T81/B062,  
3039 Cornwallis Road,  
PO Box 12195,  
Research Triangle Park, NC 27709-2195.  
Telephone: (919)

FULL NAME OF SOLE OR FIRST INVENTOR: **Alessandro Donatelli**

INVENTOR'S SIGNATURE: Alessandro Donatelli DATE: 4/11/2000  
RESIDENCE: Via della Magliana Nuova 150, Rome, Italy 00146.  
CITIZENSHIP: Italian  
POST OFFICE ADDRESS: As above

FULL NAME OF SECOND INVENTOR: **Fabrizio Loppini**

INVENTOR'S SIGNATURE: Fabrizio Loppini DATE: 4/11/2000  
RESIDENCE: Via Ascanio Vitozzi 60, Rome, Italy 00144.  
CITIZENSHIP: Italian  
POST OFFICE ADDRESS: As above

FULL NAME OF THIRD INVENTOR: **Mattia de Rosa**

INVENTOR'S SIGNATURE: Mattia De Rosa DATE: 4/11/2000  
RESIDENCE: Via Alessandro Bonci 19, Rome, Italy 00168.  
CITIZENSHIP: Italian  
POST OFFICE ADDRESS: As above

INVENTOR'S SIGNATURE: Francesca Roca DATE: 4/11/2008

CITIZENSHIP: Italian

POST OFFICE ADDRESS: As above

[illegible]

